Exhibit 4

U.S. Patent No. 7,325,733 ("'733 Patent")

Accused Products

Dell Wi-Fi enabled products including the Windows 10 operating system, including without limitation the Dell XPS 13 7390 ("Accused Products") infringe at least Claims 1, 13, and 22 of the '733 Patent.

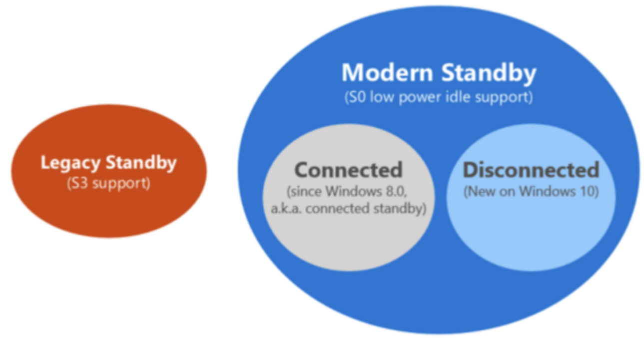
Claim 1

Claim 1	Accused Products
[1pre]. A system, comprising:	To the extent the preamble is limiting, each Accused Product comprises the claimed system. For example, the XPS 13 7390 includes the Windows 10 operating system and practices the Windows 10 "Modern Standby" feature, as described below. *See, e.g.*:

Claim 1	Accused Products
	Operating System **Available with Windows 10 Home –** Get the best combination of Windows features you know and new improvements you'll love.  Screenshot from https://www.dell.com/en-us/shop/dell-laptops/xps-13-7390-laptop/spd/xps-13-7390-laptop

Claim 1	Accused Products
	 Photograph of XPS 13 7390 device from https://www.dell.com/en-us/shop/dell-laptops/xps-13-7390-laptop/spd/xps-13-7390-laptop

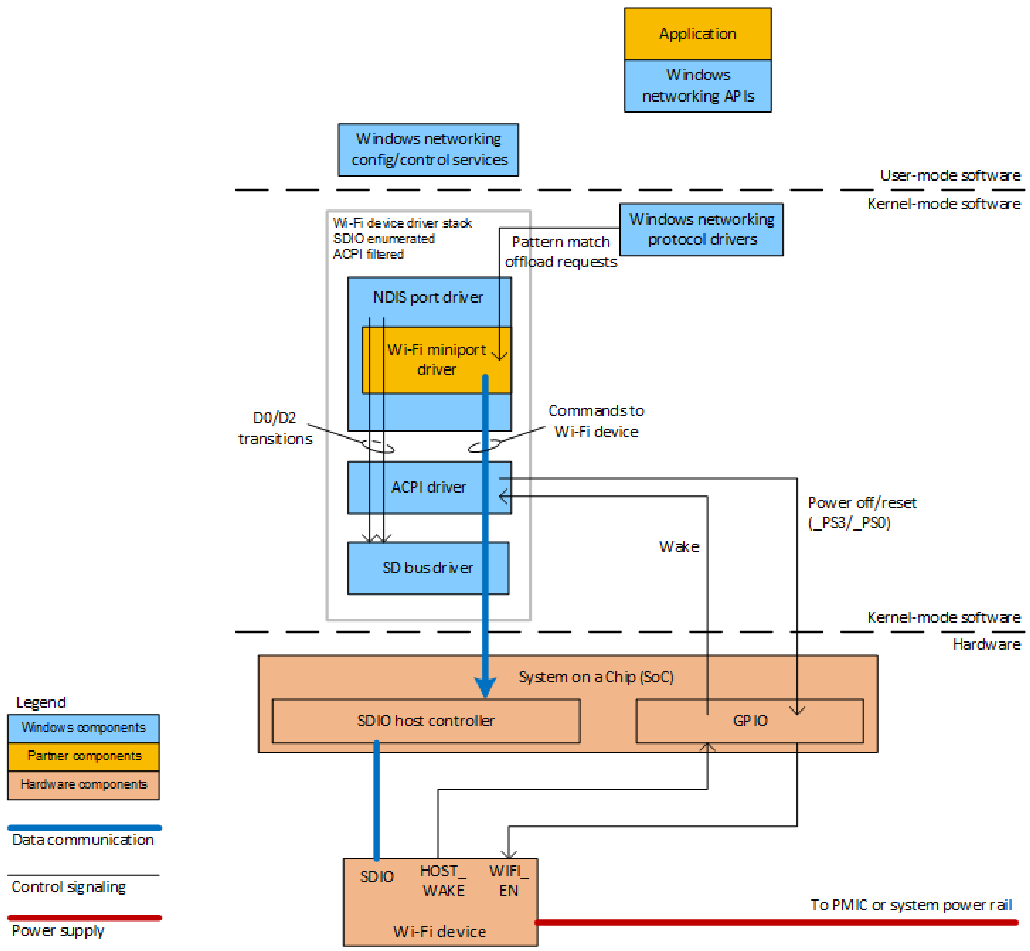
Claim 1	Accused Products
	What is Modern Standby? 05/02/2017 • 4 minutes to read • +1 Windows 10 Modern Standby (Modern Standby) expands the Windows 8.1 Connected Standby power model. Connected Standby, and consequently Modern Standby, enable an instant on / instant off user experience, similar to smartphone power models. Just like the phone, the S0 low power idle model enables the system to stay up-to-date whenever a suitable network is available. Although Modern Standby enables an instant on/off user experience like Connected Standby, Modern Standby is more inclusive than the Windows 8.1 Connected Standby power model. Modern Standby allows for market segments previously limited to the S3 power model to take advantage of the low power idle model. Example systems include systems based on rotational media and hybrid media (for example, SSD + HDD or SSHD) and/or a NIC that doesn't support all of the prior requirements for Connected Standby. The number of systems capable of S0 low power idle model is increasing and we expect more systems to use the always on, instantly available power model instead of the traditional S3/S4 power model. In the Modern Standby section, we have outlined important changes, partner requirements, and best practices for enabling Modern Standby. **Note** Modern Standby is available for both Windows 10 desktop and Windows 10 Mobile. Excerpt describing Windows Modern Standby from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby?redirectedfrom=MSDN

Claim 1	Accused Products
	 The above figure illustrates the relationships between the models and how Modern Standby systems can either stay connected to or disconnect from the network while in standby. On any Modern Standby system, the system remains in S0 while in standby, allowing the following scenarios to work: • Background activity • Faster resume from a low power state On systems that can stay connected while in standby, wakes based on specific network patterns may also be set by the operating system to enable apps to receive the latest content such as incoming email, VoIP calls, or news articles. **Resume from Modern Standby** When the user causes the system to resume from standby, e.g. presses the power button, the display is immediately turned on and networking devices are restored to their normal, active operating modes. The time from the power button press to the turning on of the display is typically less than 500 milliseconds. After the display is turned on and the networking device returns to normal operating mode, desktop applications resume and the system returns to its normal, screen-on active behavior.

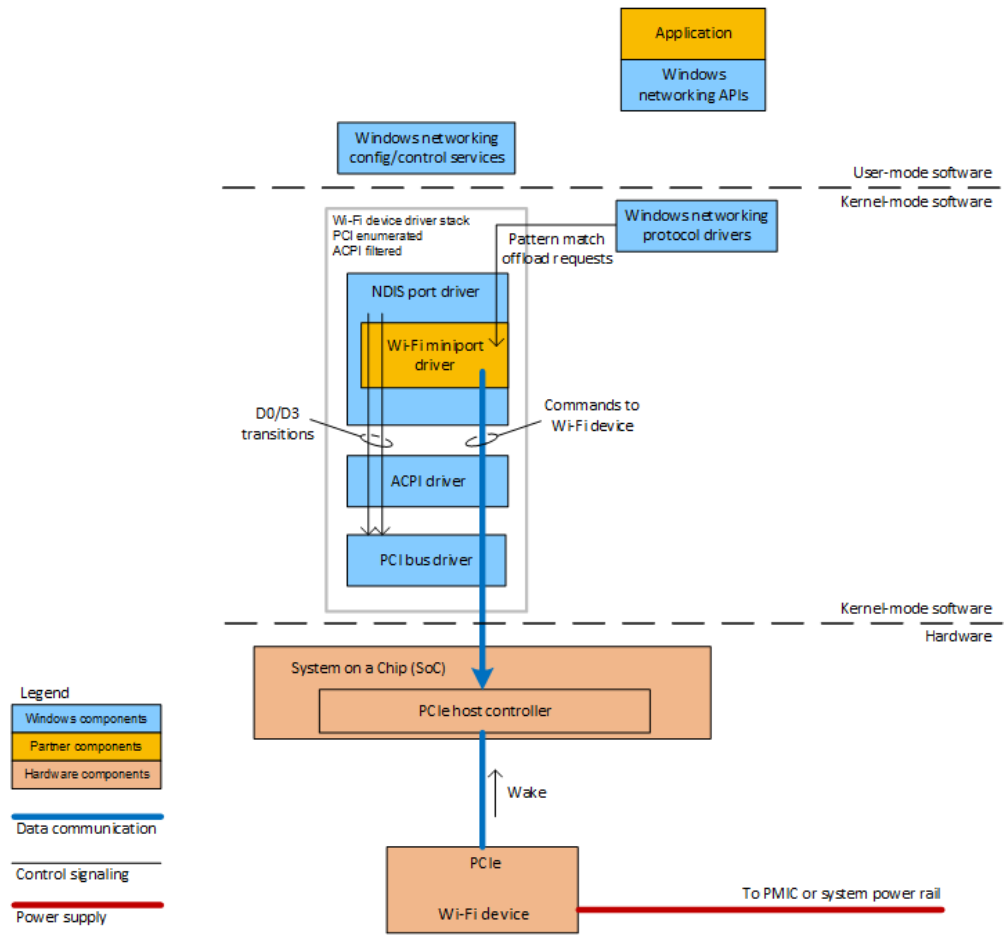
Claim 1	Accused Products
	Activity during Modern Standby On-demand transitions to active mode can occur in response to user inputs, interrupts from networking devices and other hardware events. Windows transitions the SoC platform from active mode to idle mode after all software activity is stopped and the devices on and off the SoC chip have entered low-power states. (See Transitioning between active and idle states.) The networking and communications devices automatically transition between active and low-power modes, based on the software activity of the system during modern standby. When there are no system services or Microsoft Store app background tasks that require the network, the networking device is in the low-power, protocol offload, and WoL patterns mode. When a system service or background task requires network access, Windows automatically transitions the networking device to an active mode. **Technical differences** When in the lowest power state, systems may look very similar to systems in the S3 state—processors are powered off, memory is in self-refresh. The difference is in the path of how it enters and exits low power state. For S3 systems, the system is either active or in S3. For Modern Standby, the transition from the active to the low power state is a series of steps to lower power consumption. Components are powered down when they are not in use. So, the transition into and out of a lower power state is much quicker on a Modern Standby system than on an S3 system. This design also helps with the speed of entry and exit from Standby as it doesn't require firmware interactions. Excerpts describing Windows Modern Standby from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-vs-s3

Claim 1	Accused Products
	Transitioning from active to idle Windows transitions the SoC platform from active mode to idle mode after all software activity is stopped and the devices on and off the SoC chip have entered low-power states. The SoC transitions between idle and active modes during modern standby based on the triggers explained in the previous section. Whenever the SoC is active, Windows will aggressively attempt to return it to the idle (DRIPS) state to reduce power consumption. **Note** During modern standby, the SoC can be idle or active. The SoC will also be active when the system exits modern standby in response to user input, such as a power button press. The SoC will be put back in the idle (DRIPS) state automatically when all of the following conditions are true: • All devices outside of the SoC have been powered down. • All network and radio devices have entered their low-power state to wait for interesting network packets or wake interrupts. • All host controllers on the SoC have been powered down. • All app background tasks have completed. • All CPU and GPU activity has stopped and all CPUs are idle. Almost all modern standby power problems are related to making sure all of these five conditions have been met. Windows includes a built-in diagnostic tool called SleepStudy to help discover which of these conditions is not being met when the SoC is prevented from entering the idle (DRIPS) state. Excerpts describing Windows Modern Standby from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/transitioning-between-idle-and-active-states
[1a] a processor;	Each Accused Product includes a processor. For example, the XPS 13 7390 includes an Intel Core 10th Generation processor. *See, e.g.*:

Claim 1	Accused Products
	The most powerful 13-inch laptop in its class*: With up to new 10th Gen Intel® Quad Core processors, more cores means increased performance, even with multiple applications running. Plus, Dell Power Manager, engineered by Dell, allows users to customize their laptop acoustics (fan speeds), temperature and performance based on their preferences between quiet, ultra-performance and cool modes. For example, quiet mode is 50% quieter than the other modes. It also dynamically delivers the maximum power from your processor while monitoring and managing system temperatures. When you see the “Engineered for mobile performance” icon next to select Dell PCs, you know it was designed to keep up with your fast-paced life. Screenshot from https://www.dell.com/en-us/shop/dell-laptops/xps-13-7390-laptop/spd/xps-13-7390-laptop
[1b] a host controller coupled to the processor; and	Each Accused Product comprises a host controller coupled to the processor. For example, the XPS 13 7390 comprises a host controller within the system-on-a-chip (SoC) for communicating with other devices. For one example, any device supporting Windows 10 Modern Standby must include either a PCIe host controller, an SDIO host controller, or a proprietary SoC bus controller. *See, e.g.*:

Claim 1	Accused Products
	 Excerpt, showing connection between host controller and WiFi device, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms

<table>
<tr><th>Claim 1</th><th>Accused Products</th></tr>
<tr><td></td><td>

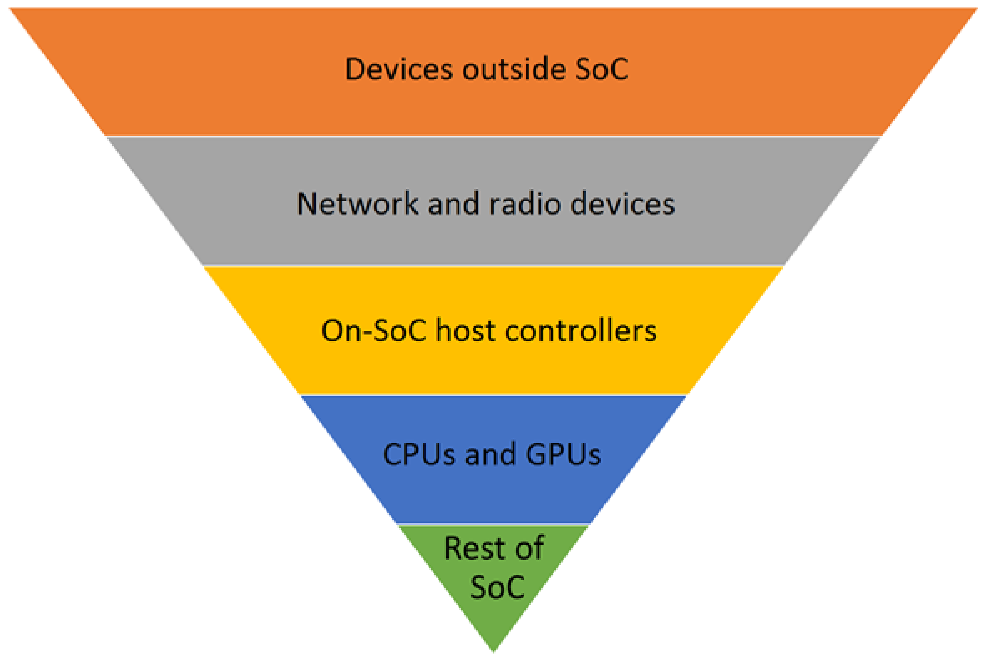


Excerpt, showing connection between host controller and WiFi device, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms

</td></tr>
</table>

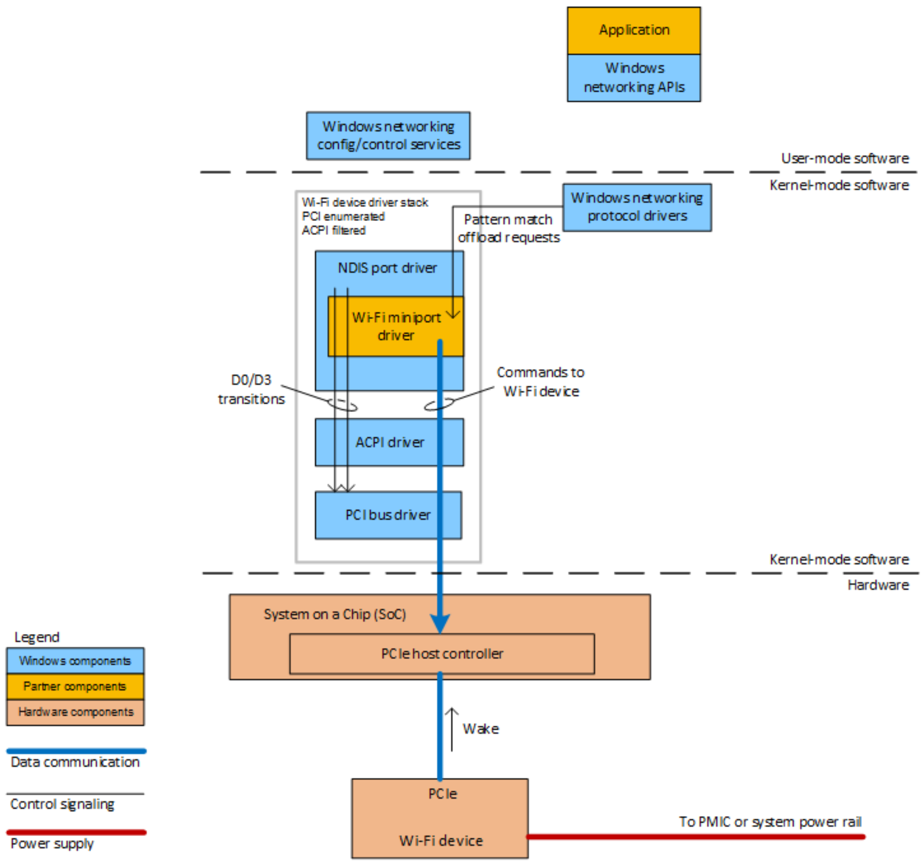
After all devices outside of the SoC, including communications devices, have been powered down, the host controllers on the SoC will be turned off. Almost every SoC has USB, I^2C, GPIO, SDIO, and UART host controllers. Each of these components on the SoC must be turned off in order for the SoC to enter a low-power mode.

The process for preparing the hardware for low-power during standby can be visualized as an upside-down pyramid, as shown in the following diagram. The lowest power is achieved when the whole SoC chip is powered down, but this can occur only after each set of devices above it in the pyramid has been powered down.



Excerpt describing connection between SOC and other devices, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/prepare-hardware-for-modern-standby

Claim 1	Accused Products
[1c] a device coupled to the host controller;	Each Accused Product includes a device coupled to the host controller. For example, the XPS 13 7390 includes a WiFi device coupled to the host controller. *See, e.g.*: **Killer™ AX1650 built on Intel WiFi 6 Chipset:** With advanced Wi-Fi 6 technology and theoretical throughput speeds of up to 2.4 Gbps, the Killer AX1650 is nearly three times as fast as the previous generation of 80MHz 2x2 AC products. It prioritizes streaming video, communication, and game traffic in your system for fast, smooth online experiences. Exceprt, describing WiFi device, from https://www.dell.com/en-us/shop/dell-laptops/xps-13-7390-laptop/spd/xps-13-7390-laptop

Claim 1	Accused Products
	 Excerpt showing connection from Wi-Fi device to host controller from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms To support modern standby, Windows is designed to use the right network adapter at the right time for an Internet

Claim 1	Accused Products
	connection. All modern standby PCs have a Wi-Fi adapter, but some PCs also have a mobile broadband (MBB) adapter and/or a wired Ethernet adapter. During modern standby, Windows automatically connects to the best available network. *** The Wi-Fi and MBB devices in a modern standby platform are expected to be highly autonomous. A key Wi-Fi device feature for modern standby is called network list offload (NLO). During modern standby, a Wi-Fi device that supports NLO can automatically connect to previously used Wi-Fi access points. NLO allows the Wi-Fi device to roam between previously used access points while the SoC remains in the low-power idle mode. As the user commutes between home and work, Windows automatically connects to Wi-Fi, thereby allowing the system to be already connected when the user presses the power button. The user no longer has to wait to connect to Wi-Fi. Instead, Wi-Fi is connected before the user turns on the system. Similarly, email is already downloaded, and connections to Skype and other communications services are automatically reconnected. Excerpt from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/networking-power-management-for-modern-standby-platforms

Claim 1	Accused Products
	A Wi-Fi device in a modern standby platform must support several key power-management features to reduce power consumption by both the device and the platform as a whole. The first feature—called power save mode—allows the Wi-Fi device to reduce its power consumption while it remains connected to the access point. Power save mode reduces power consumption at the cost of increased data transfer latency. Power save mode is expected to always be enabled when the platform is running on battery power, except when low-latency connections are required (for example, for VOIP calls). For more information, see Wi-Fi Auto Power Save Mode. The second key power-management feature is pattern-match wake. This feature allows Windows to arm the Wi-Fi device to wake the System on a Chip (SoC) when the Wi-Fi device detects a network packet that matches a stored pattern. Pattern-match wake is operational only during modern standby. While pattern-match wake is enabled, the Wi-Fi device operates in a very low-power mode and listens for incoming data destined for specific system services or registered applications (for example, push notifications and email). Meanwhile, the other components in the hardware platform are in a low-power state. For more information, see Network Wake-Up Events. In addition, Wi-Fi devices in a modern standby platform must support the following run-time power-management features:

Claim 1	Accused Products
	• Radio on/off state • Network list offload (NLO) • ARP/NS offload • D0 packet coalescing • Dynamic DTIM management • A set of Wi-Fi connectivity triggers, which include wake-on-AP-disconnect Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms
[1d] wherein the device is electrically disconnected from the host controller if the device is not in an active state; and	In each Accused Product, the device is electrically disconnected from the host controller if the device is not in an active state. For example, the device is not in an active state when the system enters Modern Standby idle mode. When the device is in this non-active state, the Windows 10 Modern Standby software electrically disconnects the device from the host controller, for example by configuring the device and/or host controller not to communicate with the SoC except for wake interrupts. *See, e.g.*:

Claim 1	Accused Products
	Transitioning from active to idle Windows transitions the SoC platform from active mode to idle mode after all software activity is stopped and the devices on and off the SoC chip have entered low-power states. The SoC transitions between idle and active modes during modern standby based on the triggers explained in the previous section. Whenever the SoC is active, Windows will aggressively attempt to return it to the idle (DRIPS) state to reduce power consumption. **Note** During modern standby, the SoC can be idle or active. The SoC will also be active when the system exits modern standby in response to user input, such as a power button press. The SoC will be put back in the idle (DRIPS) state automatically when all of the following conditions are true: • All devices outside of the SoC have been powered down. • All network and radio devices have entered their low-power state to wait for interesting network packets or wake interrupts. • All host controllers on the SoC have been powered down. • All app background tasks have completed. • All CPU and GPU activity has stopped and all CPUs are idle. Almost all modern standby power problems are related to making sure all of these five conditions have been met. Windows includes a built-in diagnostic tool called SleepStudy to help discover which of these conditions is not being met when the SoC is prevented from entering the idle (DRIPS) state. Excerpt, describing electrical disconnection, from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/transitioning-between-idle-and-active-states

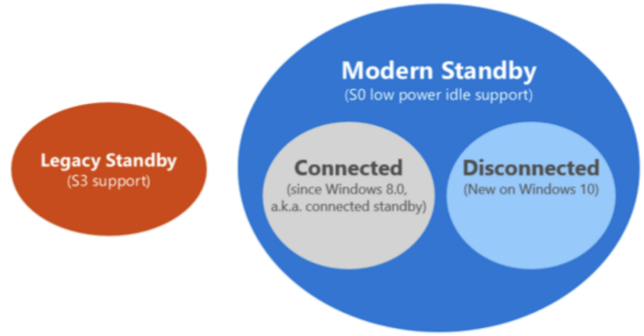
	Connected sleep The Wi-Fi device is connected to the access point, but the remainder of the platform is in a very low-power state. Pattern-match wake is enabled so that the Wi-Fi device wakes the SoC on a specific set of incoming network packets. Transition mechanism: • Before the Wi-Fi device leaves D0, NDIS will send an OID_PM_ADD_WOL_PATTERN request to instruct the Wi-Fi miniport driver to add wake-on-LAN patterns. • To instruct the Wi-Fi miniport driver to enable pattern-match wake, NDIS will send an OID_PM_PARAMETERS request. • NDIS will send an OID_PNP_SET_POWER request with an **NDIS_DEVICE_POWER_STATE** value of NdisDeviceStateD2 (for SDIO) or NdisDeviceStateD3 (for PCIe). *Disconnected sleep* The Wi-Fi device is powered but is not connected to an access point, because no preferred access point is within range. The remainder of the platform is in a very low-power state. Pattern-match wake is enabled and the Network Offload List is plumbed to the Wi-Fi device. The Wi-Fi device uses the Network Offload List to periodically scan for preferred networks to connect to. Transition mechanism: • The Wi-Fi device uses the network offload list to periodically scan for preferred networks to connect to. • If a matching network is found during these periodic scans, the Wi-Fi device will wake the SoC.

	Radio off The Wi-Fi device still has power applied, but the radio (RF components) has been powered down. Transition mechanism: • While in D0, NDIS will send an OID_DOT11_NIC_POWER_STATE request with a value of FALSE, indicating the radio should be powered off. *Device powered down* The Wi-Fi device has been completely powered down. Transition mechanism: • NDIS will send an OID_PNP_SET_POWER request with an **NDIS_DEVICE_POWER_STATE** value of NdisDeviceStateD3. • If the Wi-Fi device is connected to SDIO or PCIe, the system ACPI firmware will remove power from or reset the Wi-Fi device by using a GPIO line from the SoC to the Wi-Fi device. • If the Wi-Fi device is integrated into the SoC, the system firmware is responsible for powering off or resetting the Wi-Fi device by using a proprietary mechanism.

Claim 1	Accused Products
	Pattern-match wake Windows requires all networking devices in a modern standby platform to support pattern-match wake. This feature enables the network device to monitor the network connection for interesting packets (by comparing incoming packets to patterns stored in the device) while the SoC or chipset is powered off, and to wake the main processor on the SoC or chipset when a matching pattern is detected. This ability to offload Wi-Fi processing from the main processor is key to achieving low-power operation in modern standby. Before the platform enters modern standby and the Wi-Fi device transitions to the connected-sleep (D2 for SDIO; D3 for PCIe) mode, Windows sends the Wi-Fi miniport driver a collection of interesting patterns that the Wi-Fi device must watch for. Pattern-match wake is the key enabling feature for modern standby. Pattern-match wake is enabled when the Wi-Fi device is in connected-sleep (D2/D3) mode. Before the Wi-Fi device enters this mode, Windows instructs the device to ignore all network packets except for those that match a specified set of patterns. Windows builds the matching patterns based on the services and applications (for example, push notifications and email) that are currently in use. Offloading pattern matching to the Wi-Fi device allows the rest of the hardware platform either to be turned off or to operate in a low-power mode. Meanwhile, the Wi-Fi device stays turned on, waiting for important incoming network traffic. Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms

Claim 1	Accused Products
	Connected vs. disconnected standby Connected and disconnected Modern Standby states transitions follow a very similar code flow. The biggest difference is maintaining network connection. When a system goes into disconnected Standby state, it enters DRIPS, just like with the connected state. The primary difference is that it stops Wi-Fi traffic if a NIC that is not capable of offload support is detected. When a system enters disconnected standby, no patterns are configured for wake across LAN, Wi-Fi, or Mobile broadband. A media disconnect is sent to LAN, Wi-Fi and Mobile Broadband parts. Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-network-connectivity
[1e] wherein the device being electrically disconnected from the host controller causes an appearance to the host controller that the device is not coupled to the host controller.	In each Accused Product, the device being electrically disconnected from the host controller causes an appearance to the host controller that the device is not coupled to the host controller. For example, under the Windows S0 low-power idle mode ("Modern Standby"), the host controller indicates that the network is disconnected. *See, e.g.*:

Claim 1	Accused Products
	Technical differences When in the lowest power state, systems may look very similar to systems in the S3 state—processors are powered off and memory is in self-refresh. The difference is in the path of how it enters and exits low power state. For S3 systems, the system is either active or in S3. For Modern Standby, the transition from the active to the low power state is a series of steps to lower power consumption. Components are powered down when they are not in use. So, the transition into and out of a lower power state is much quicker on a Modern Standby system than on an S3 system. This design also helps with the speed of entry and exit from Standby as it doesn't require firmware interactions. Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-vs-s3

Claim 1	Accused Products
	 The above figure illustrates the relationships between the models and how Modern Standby systems can either stay connected to or disconnect from the network while in standby. On any Modern Standby system, the system remains in S0 while in standby, allowing the following scenarios to work: • Background activity • Faster resume from a low power state On systems that can stay connected while in standby, wakes based on specific network patterns may also be set by the operating system to enable apps to receive the latest content such as incoming email, VoIP calls, or news articles. Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-vs-s3

Claim 1	Accused Products
	The Modern Standby user experience is designed to model that of a cellular phone. When users finish using their phones, they press the system power button and the cell phone enters sleep mode. The phone remains asleep until the user presses the power button again, or a phone call, email, or instant message is received. Similarly, when a PC is in Modern Standby, it looks and feels off—the screen is blanked, the system has no visible LED indicators, and there is no acoustic noise. However, a PC in Modern Standby remains on and connected to the Internet, just as the cell phone remains connected to the cellular network. (The Modern Standby PC uses any available network connection—Wi-Fi, mobile broadband (MBB)/cellular, or wired Ethernet.) And the Modern Standby PC, connected or not, also has very long battery life in its screen-off state, just like a cell phone. Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-wake-sources

Claim 13

Claim 13	Accused Products
[13pre]. A method, comprising:	To the extent the preamble is limiting, each Accused Product performs the claimed method. *See supra* claim element [1pre].
[13a] detecting whether a device coupled to a host controller is in an active state;	Each Accused Product performs detecting whether a device coupled to a host controller is in an active state. For example, the device is detected to be in an active state when the system is not in Modern Standby idle mode. *See*, *e.g.*:

Claim 13	Accused Products
	Transitioning from active to idle Windows transitions the SoC platform from active mode to idle mode after all software activity is stopped and the devices on and off the SoC chip have entered low-power states. The SoC transitions between idle and active modes during modern standby based on the triggers explained in the previous section. Whenever the SoC is active, Windows will aggressively attempt to return it to the idle (DRIPS) state to reduce power consumption. **Note** During modern standby, the SoC can be idle or active. The SoC will also be active when the system exits modern standby in response to user input, such as a power button press. The SoC will be put back in the idle (DRIPS) state automatically when all of the following conditions are true: • All devices outside of the SoC have been powered down. • All network and radio devices have entered their low-power state to wait for interesting network packets or wake interrupts. • All host controllers on the SoC have been powered down. • All app background tasks have completed. • All CPU and GPU activity has stopped and all CPUs are idle. Almost all modern standby power problems are related to making sure all of these five conditions have been met. Windows includes a built-in diagnostic tool called SleepStudy to help discover which of these conditions is not being met when the SoC is prevented from entering the idle (DRIPS) state. Excerpt from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/transitioning-between-idle-and-active-states

	Connected sleep The Wi-Fi device is connected to the access point, but the remainder of the platform is in a very low-power state. Pattern-match wake is enabled so that the Wi-Fi device wakes the SoC on a specific set of incoming network packets. Transition mechanism: • Before the Wi-Fi device leaves D0, NDIS will send an OID_PM_ADD_WOL_PATTERN request to instruct the Wi-Fi miniport driver to add wake-on-LAN patterns. • To instruct the Wi-Fi miniport driver to enable pattern-match wake, NDIS will send an OID_PM_PARAMETERS request. • NDIS will send an OID_PNP_SET_POWER request with an **NDIS_DEVICE_POWER_STATE** value of NdisDeviceStateD2 (for SDIO) or NdisDeviceStateD3 (for PCIe). *Disconnected sleep* The Wi-Fi device is powered but is not connected to an access point, because no preferred access point is within range. The remainder of the platform is in a very low-power state. Pattern-match wake is enabled and the Network Offload List is plumbed to the Wi-Fi device. The Wi-Fi device uses the Network Offload List to periodically scan for preferred networks to connect to. Transition mechanism: • The Wi-Fi device uses the network offload list to periodically scan for preferred networks to connect to. • If a matching network is found during these periodic scans, the Wi-Fi device will wake the SoC.

	Radio off The Wi-Fi device still has power applied, but the radio (RF components) has been powered down. Transition mechanism: • While in D0, NDIS will send an OID_DOT11_NIC_POWER_STATE request with a value of FALSE, indicating the radio should be powered off. *Device powered down* The Wi-Fi device has been completely powered down. Transition mechanism: • NDIS will send an OID_PNP_SET_POWER request with an **NDIS_DEVICE_POWER_STATE** value of NdisDeviceStateD3. • If the Wi-Fi device is connected to SDIO or PCIe, the system ACPI firmware will remove power from or reset the Wi-Fi device by using a GPIO line from the SoC to the Wi-Fi device. • If the Wi-Fi device is integrated into the SoC, the system firmware is responsible for powering off or resetting the Wi-Fi device by using a proprietary mechanism.

Claim 13	Accused Products
	Pattern-match wake Windows requires all networking devices in a modern standby platform to support pattern-match wake. This feature enables the network device to monitor the network connection for interesting packets (by comparing incoming packets to patterns stored in the device) while the SoC or chipset is powered off, and to wake the main processor on the SoC or chipset when a matching pattern is detected. This ability to offload Wi-Fi processing from the main processor is key to achieving low-power operation in modern standby. Before the platform enters modern standby and the Wi-Fi device transitions to the connected-sleep (D2 for SDIO; D3 for PCIe) mode, Windows sends the Wi-Fi miniport driver a collection of interesting patterns that the Wi-Fi device must watch for. Pattern-match wake is the key enabling feature for modern standby. Pattern-match wake is enabled when the Wi-Fi device is in connected-sleep (D2/D3) mode. Before the Wi-Fi device enters this mode, Windows instructs the device to ignore all network packets except for those that match a specified set of patterns. Windows builds the matching patterns based on the services and applications (for example, push notifications and email) that are currently in use. Offloading pattern matching to the Wi-Fi device allows the rest of the hardware platform either to be turned off or to operate in a low-power mode. Meanwhile, the Wi-Fi device stays turned on, waiting for important incoming network traffic. Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/wi-fi-power-management-for-modern-standby-platforms

Claim 13	Accused Products
	Connected vs. disconnected standby Connected and disconnected Modern Standby states transitions follow a very similar code flow. The biggest difference is maintaining network connection. When a system goes into disconnected Standby state, it enters DRIPS, just like with the connected state. The primary difference is that it stops Wi-Fi traffic if a NIC that is not capable of offload support is detected. When a system enters disconnected standby, no patterns are configured for wake across LAN, Wi-Fi, or Mobile broadband. A media disconnect is sent to LAN, Wi-Fi and Mobile Broadband parts. Excerpts from https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-network-connectivity
[13b] if the device is not in an active state, electrically disconnecting the device from a host controller, wherein electrically disconnecting the device from the host controller causes an appearance to the host controller that the device is not coupled to the host controller; and	Each Accused Product performs if the device is not in an active state, electrically disconnecting the device from a host controller, wherein electrically disconnecting the device from the host controller causes an appearance to the host controller that the device is not coupled to the host controller. *See supra* claim elements [1d] and [1e].
[13c] if the device is in an active state, maintaining an electrical connection between the device and the host controller.	Each Accused Product performs if the device is in an active state, maintaining an electrical connection between the device and the host controller. For example, when the system is not in Modern Standby idle mode, the system does not electrically disconnect the device from the host controller. *See supra* claim element [1d].

Claim 22

Claim 22	Accused Products
[22pre]. A computer accessible memory medium that stores program instructions, wherein the program instructions are executable by a processor to:	Each Accused Product includes a computer accessible memory medium that stores program instructions, wherein the program instructions are executable by a processor to perform the claimed steps. *See supra* claim element [1pre].
[22a] detect whether a device coupled to a host controller is in an active state;	In each Accused Product, the program instructions are executable to detect whether a device coupled to a host controller is in an active state. *See supra* claim element [13a].
[22b] if the device is not in an active state, electrically disconnect the device from a host controller, wherein electrically disconnecting the device from the host controller causes an appearance to the host controller that a device is not coupled to the host controller; and	In each Accused Product, the program instructions are executable to, if the device is not in an active state, electrically disconnect the device from a host controller, wherein electrically disconnecting the device from the host controller causes an appearance to the host controller that a device is not coupled to the host controller. *See supra* claim elements [1d] and [1e].
[22c] if the device is in an active state, maintain an electrical connection between the device and the host controller.	In each Accused Product, the program instructions are executable to, if the device is in an active state, maintain an electrical connection between the device and the host controller. *See supra* claim element [13c].